

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra kybernetiky a biomedicínského inženýrství

# **Návrh a realizace snímače prašnosti s bezdrátovým přenosem dat**

## **Design and Implementation of a Dustiness Sensor with Wireless Data Transmission**

## Zadání bakalářské práce

Student: **Jakub Klokočník**

Studijní program: **B2649 Elektrotechnika**

Studijní obor: **2612R041 Řídící a informační systémy**

Téma: **Návrh a realizace snímače prašnosti s bezdrátovým přenosem dat**  
**Design and Implementation of a Dustiness Sensor with Wireless**  
**Data Transmission**

Jazyk vypracování: **čeština**

### Zásady pro vypracování:

1. Seznámení se s technologií IQRF.
2. Seznámení se se snímačem prašnosti Honeywell HPM115S0-XXX a jeho komunikačním protokolem.
3. Návrh a realizace bezdrátového měřicího uzlu pro měření prašnosti využívajícím technologii IQRF pro komunikaci.
4. Návrh a realizace vizualizace měřených dat prašnosti ve vhodném vizualizačním prostředí.
5. Zhodnocení výsledků a závěr.

### Seznam doporučené odborné literatury:

- [1] JACKO, Róbert. Energeticky soběstačný bezdrátový modul pro senzorické aplikace [online]. Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií, 2014 [cit. 2018-10-29]. Dostupné z: <http://hdl.handle.net/11012/32913>. Diplomová práce. Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií. Ústav telekomunikací.
- [2] BRTNÍK, Bohumil a David MATOUŠEK. *Algoritmy číslicového zpracování signálů*. Praha: BEN - technická literatura, 2011. ISBN 978-80-7300-400-2.
- [3] MICRORISC s.r.o. *Complete technology for MESH networks*. [online]. [cit.2018-10-01]. c2013. Dostupné z: <http://iqr.org/weben/index.php>.
- [4] VRBA, Kamil, et al. *Konstrukce elektronických zařízení*. Brno: Vysoké učení technické v Brně, 2013. s. 251. ISBN 978-80-214-4719-6. Dostupné také z: [http://homen.vsb.cz/~hav278/Konstrukce\\_zarizeni\\_prumyslove\\_elektroniky/Studijni\\_opory/Konstrukce\\_elektronickych\\_zarizeni.pdf](http://homen.vsb.cz/~hav278/Konstrukce_zarizeni_prumyslove_elektroniky/Studijni_opory/Konstrukce_elektronickych_zarizeni.pdf).
- [5] MICRORISC s.r.o. *IQRF DPA Framework Technical Guide v3.02*. 2018. [online] Dostupné z: <http://www.iqr.org>.
- [6] HONEYWELL. *HPM115S0-XXX Particle Sensors*. 2018. [online] Dostupné z: <https://sensing.honeywell.com/honeywell-sensing-hpm-series-particle-sensors-datasheet-32322550-e-en.pdf>

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Martin Pleš, Ph.D.**

Datum zadání: 01.09.2018

Datum odevzdání: 30.04.2019



doc. Ing. Jiří Koziorek, Ph.D.  
*vedoucí katedry*



prof. Ing. Pavel Brandštetter, CSc.  
*děkan fakulty*

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 23. dubna 2019




.....



Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava.

V Ostravě 23. dubna 2019



.....

Rád bych poděkoval panu Ing. Martinu Piešovi, Ph.D. za vstřícnost, odbornou pomoc a cenné rady při konzultacích a vytváření této bakalářské práce.

## Abstrakt

Tato bakalářská práce se zabývá návrhem a realizací snímače prašnosti s bezdrátovým přenosem dat. Výsledné zařízení je uzpůsobeno pro minimální náročnost na spotřebu elektrické energie. Pro bezdrátový přenos naměřených dat ze snímače prašnosti je využita technologie IQRf. Řídicí program je realizován ve vizualizačním prostředí Node-RED, který běží na platformě Raspberry Pi. V první části bakalářské práce jsou nejprve teoreticky popsány využívané technologie, v druhé části je rozebrán postup návrhu a realizace elektroniky, krabičky, komunikace a vizualizace. Výsledné zařízení je umístěno do krabičky vytištěné na 3D tiskárně a bude sloužit jako měřicí uzel znečištění ovzduší ve vnitřních prostorech školy.

**Klíčová slova:** Snímač prašnosti, Honeywell, HPM115S0, IQRf, TR-76DA, MESH, DPS, DPA, modul, C, nod, koordinátor, MQTT,  $\text{\LaTeX}$

## Abstract

This bachelor thesis contains design and implementation of a dustiness sensor with wireless data transmission. The final device is designed to minimize power consumption. IQRf technology is used for wireless transmission of measured data from the dust sensor. The control program is implemented in the Node-RED visualization environment, which runs on Raspberry Pi platform. In the first part of the thesis there are theoretical bases that relate to theoretical knowledge and procedures. In the second part is analysed procedure of design and realization of electronics, box, communication and visualization. The final device is placed in a box, which was printed on a 3D printer and its purpose is measuring air pollution inside of the school.

**Key Words:** Dustiness sensor, Honeywell, HPM115S0, IQRf, TR-76DA, MESH, DPS, DPA, module, C, node, coordinator, MQTT,  $\text{\LaTeX}$

# Obsah

Seznam použitých zkratk a symbolů	9
Seznam obrázků	10
Seznam tabulek	11
<b>1 Úvod</b>	<b>12</b>
<b>2 Technologie IQRF</b>	<b>13</b>
2.1 Vývojový kit DS-START-04 . . . . .	14
2.2 Komunikace . . . . .	15
2.3 Způsoby programování . . . . .	16
2.4 Softwarové prostředky . . . . .	18
2.5 IQRF GW Daemon . . . . .	19
<b>3 Platforma Raspberry Pi</b>	<b>20</b>
<b>4 Snímač prašnosti Honeywell HPMA115S0</b>	<b>21</b>
4.1 Princip . . . . .	21
4.2 Komunikace - UART . . . . .	22
<b>5 Testování</b>	<b>25</b>
<b>6 Návrh a realizace</b>	<b>28</b>
6.1 Měřicí uzel . . . . .	28
6.2 Řídicí uzel . . . . .	34
<b>7 Vizualizace</b>	<b>37</b>
<b>8 Zhodnocení výsledků</b>	<b>40</b>
<b>9 Závěr</b>	<b>42</b>
<b>Literatura</b>	<b>43</b>
<b>Přílohy</b>	<b>43</b>
<b>A Seznam příloh</b>	<b>44</b>

## Seznam použitých zkratk a symbolů

Bajt	– Informace složená z 8 bitů
Bit	– Nejmenší dále nedělitelná jednotka informace
CPU	– Central processing unit
DPA	– Direct Peripheral Access – Komunikační protokol IQMESH sítí
DPS	– Deska plošných spojů
Filament	– Tisková struna určitého materiálu
FRC	– Fast Response Command
Gateway	– Brána, která slouží pro připojení k internetu
GW	– Gateway - brána
HWP	– Hardware profiles – Hardwarový profil
IDE	– Integrated development environment – vývojové prostředí
IQRF	– Bezdrátová technologie pro přenos malého objemu dat
IoT	– Internet of Things - "Internet věcí"
I/O	– Input/Output – Vstup/Výstup
JSON	– Javascriptový objektový zápis
Koordinátor	– Řídící prvek mesh sítě
LED	– Light emitting diode – Světlo emitující dioda
LiPol	– Lithium Polymerový akumulátor
LDO	– Low-dropout regulator
LP	– Low power
MQTT	– Message Queuing Telemetry Transport
Node	– Uzel mesh sítě
OS	– Operační systém
Paritní bit	– Kontrolní bit, určuje počet jedničkových bitů ve slově
PLA	– Polylactic acid – polymléčná kyselina, biologicky rozložitelný polyesterový výrobek z rostlinných materiálů
RAM	– Random access memory
Routování	– Určení cesty datagramu
RPI	– Jednodeskový počítač Raspberry Pi
SMT	– Surface mount technology - povrchová montáž součástek
SPI	– Serial Peripheral Interface
Start bit	– Bit, který zahajuje začátek přenosu bajtu
STD	– Standard
Stop bit	– Bit, který ukončuje přenos bajtu
UART	– Universal asynchronous receiver transceiver
USB	– Universal serial bus – Univerzální sériová sběrnice

## Seznam obrázků

1	Transceiver TR-72D . . . . .	13
2	Vývojový kit DS-START-04A [1] . . . . .	14
3	Zjednodušené schéma DK-EVAL-04A [1] . . . . .	15
4	Zjednodušené schéma CK-USB-04A . . . . .	16
5	Způsob programování - využití DPA . . . . .	17
6	Způsob programování - přístup přes OS . . . . .	18
7	Přehled funkcionalit IQRf GW Daemona . . . . .	19
8	Platforma Raspberry Pi 3 model B . . . . .	20
9	Snímač prašnosti Honeywell HPM115S0 . . . . .	21
10	Struktura snímače . . . . .	22
11	Blokové schéma zapojení . . . . .	25
12	Vývojový diagram softwaru pro testování na vývojové desce Arduino . . . . .	26
13	Výpis naměřených dat do terminálu . . . . .	27
14	Schéma IQMESH sítě . . . . .	28
15	Zjednodušené blokové elektrické schéma . . . . .	29
16	Návrh krabičky v Autodesk Inventor 2019 . . . . .	30
17	Vytištěná krabička . . . . .	30
18	Sestavená tiskárna Prusa i3 mk3 . . . . .	31
19	Vývojový diagram firmwaru . . . . .	32
20	Realizace řídicího uzlu . . . . .	34
21	Princip fungování MQTT . . . . .	36
22	Front-end vizualizace . . . . .	37
23	Back-end vizualizace . . . . .	39
24	Měřicí uzel - vnitřní pohled . . . . .	40

## Seznam tabulek

1	Struktura příkazu pro čtení ze snímače [3] . . . . .	23
2	Struktura datového protokolu [3] . . . . .	24
3	Struktura DPA požadavku z řídicího uzlu - požadavek na uživatelskou periferii .	33
4	Struktura DPA odpovědi z měřicího uzlu . . . . .	33
5	Struktura DPA odpovědi z měřicího uzlu, uživatelská data . . . . .	33

# 1 Úvod

Účelem této bakalářské práce bylo navržení a vytvoření funkčního zařízení pro měření znečištění ovzduší. Pro měření kvality ovzduší byl použit snímač HPM115S0 od firmy Honeywell, bezdrátový přenos naměřených dat byl realizován pomocí technologie IQRf a modulu TR-76D, který snímači posílá příkazy k zahájení a ukončení měření, komunikace probíhá přes sériové rozhraní UART. Tato naměřená data jsou modulem neboli uzlem sítě posílána koordinátorovi sítě, který je připojen k jednodeskovému počítači Raspberry Pi, tento počítač slouží jako internetová brána a zároveň naměřená data vizualizuje ve vizualizačním nástroji Node-RED.

V první kapitole je detailněji popsána bezdrátová technologie IQRf, způsoby jakými mohou zařízení komunikovat, možné způsoby programování, hlavní hardwarové prostředky, které byly pro realizaci praktické části práce použity a použité softwarové prostředky. Náplní druhé kapitoly bude seznámení se s platformou Raspberry Pi, která bude fungovat jako internetová brána a bude na ní realizována vizualizace. Poslední teoretickou částí bude popis a způsob komunikace snímače prašnosti Honeywell HPM115S0.

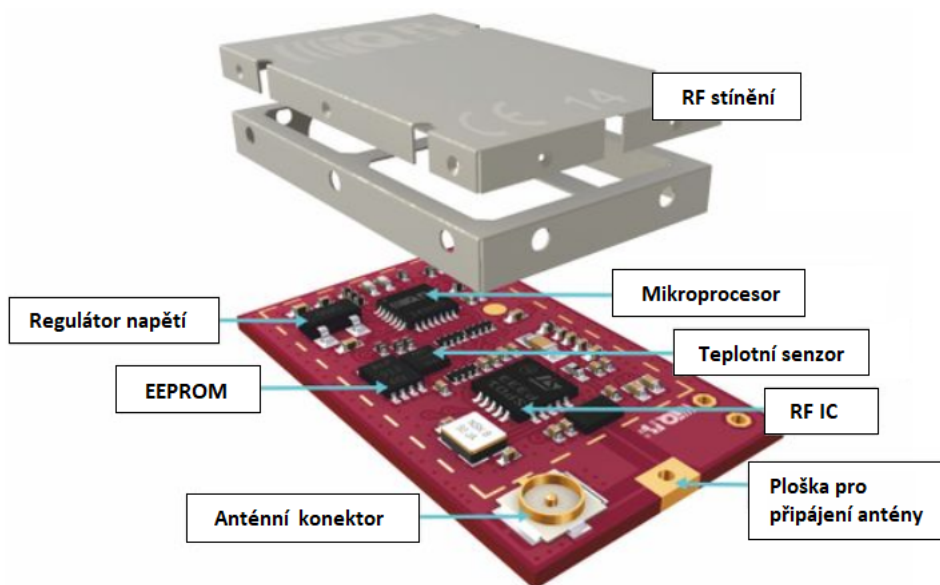
V praktické části práce bude snímač prašnosti nejprve otestován na malém jednodeskovém počítači Arduino, založeném na mikrokontroléru Atmel ve vývojovém prostředí Arduino IDE a programovacím jazyku Wiring. Následovat bude návrh elektrického schématu, desky plošných spojů měřicího uzlu a krabičky s následnou realizací. Krabička bude uzpůsobena v závislosti na rozměrech výsledného návrhu DPS a ostatních součástí. Firmware měřicího uzlu bude napsán v jazyce C za pomoci DPA frameworku. Nejprve bude otestována funkčnost celého zařízení v klasickém režimu se standardním hardwarovým profilem a poté bude firmware přizpůsoben energeticky méně náročnému hardwarovému profilu. Druhým zařízením, které bude tato práce obsahovat, je řídicí uzel, který vznikne kombinací technologie IQRf, platformy Raspberry Pi, komunikačního protokolu MQTT a vizualizačním nástrojem Node-RED. Závěr práce bude věnován zhodnocení výsledků a možným vylepšením stávajícího zařízení.



## 2 Technologie IQRF

IQRF je technologie zabývající se bezdrátovým přenosem maloobjemových dat představená v roce 2004 českou firmou MICRORISC s.r.o. a zahrnuje všechny potřebné hardwarové a softwarové prostředky pro bezdrátový přenos dat. Tato technologie je zejména vhodná pro aplikace v oblasti IoT a měření. V roce 2017 se z firmy MICRORISC oddělila část firmy a vytvořila novou firmu IQRF Tech s.r.o., která přebrala veškeré záležitosti týkající se IQRF technologie. [1]

Pro přenos dat využívá volné frekvenční pásmo 868, 433 a 916 MHz. Mezi hlavní výhody této technologie patří rychlá a pohodlná implementace do nových nebo již vytvořených systémů, z důvodu bezdrátového přenosu i praktičnost a relativně velký dosah, který je ve volném prostoru až několik stovek metrů, v budovách je dosah znatelně menší, avšak díky protokolu IQMESH a propojení několika uzlů, můžeme snadno tento dosah zvýšit. Podmínka propojitelnosti jednotlivých uzlů je ta, že jednotlivé uzly musí být v dosahu, aby mohly komunikovat, což se dá zařídit softwarovým zvýšením vysílacího výkonu. Mezi další výhody IQRF technologie patří nízká energetická náročnost. Spotřeba elektrického proudu v přijímacím režimu se standardním hardwarovým profilem je přibližně 12 mA, v klidovém režimu okolo 1,6 mA a ve vysílacím režimu až 19 mA. Při použití LP hardwarového profilu je spotřeba proudu maximálně 12 mA v režimu vysílání. Spotřeba záleží na nastaveném vysílacím výkonu. [1]



Obrázek 1: Transceiver TR-72D

Vytvořená MESH síť může obsahovat až 239 uzlů a musí mít minimálně jednoho koordinátora sítě, který posílá požadavky uzlům. Tyto uzly mohou být jakkoliv propojeny, je však vhodné mít vytvořenou takovou síť, kde existují alespoň dvě možná spojení mezi koordinátorem sítě a nejvzdálenějším uzlem z důvodu možnosti výpadku některého uzlu. Při návrhu musí být zohledněn i fakt, že předávání požadavků mezi jednotlivými uzly může trvat až 50 ms.

Relativní nevýhodou může být nízká přenosová rychlost a malý objem dat, proto je tato technologie vhodná spíše pro oblast měření a IoT. Maximální rychlost přenosu je 19,836 kb/s a data jsou posílána v paketech o maximální velikosti 64 B. Využívání této technologie nepodléhá žádným licenčním poplatkům. [1]

Hlavním hardwarovým prostředkem jsou transceivery, které jsou vyráběny v několika provedeních. Tyto transceivery se velikostně přirovnávají k velikosti SIM karty a slouží k přijímání a odesílání dat pomocí rádiových vln. Programují se za pomoci speciálního programátoru, který je součástí vývojové sady DS-START-04, která bude popsána v další podkapitole.

## 2.1 Vývojový kit DS-START-04

Univerzální vývojový kit pro IQRF transceivery. Součástí tohoto vývojového kitu jsou tři transceivery TR-72DA, dva node kity DK-EVAL-04A, programátor CK-USB-04A, USB micro kabel a flash disk s potřebnou dokumentací a softwarem.

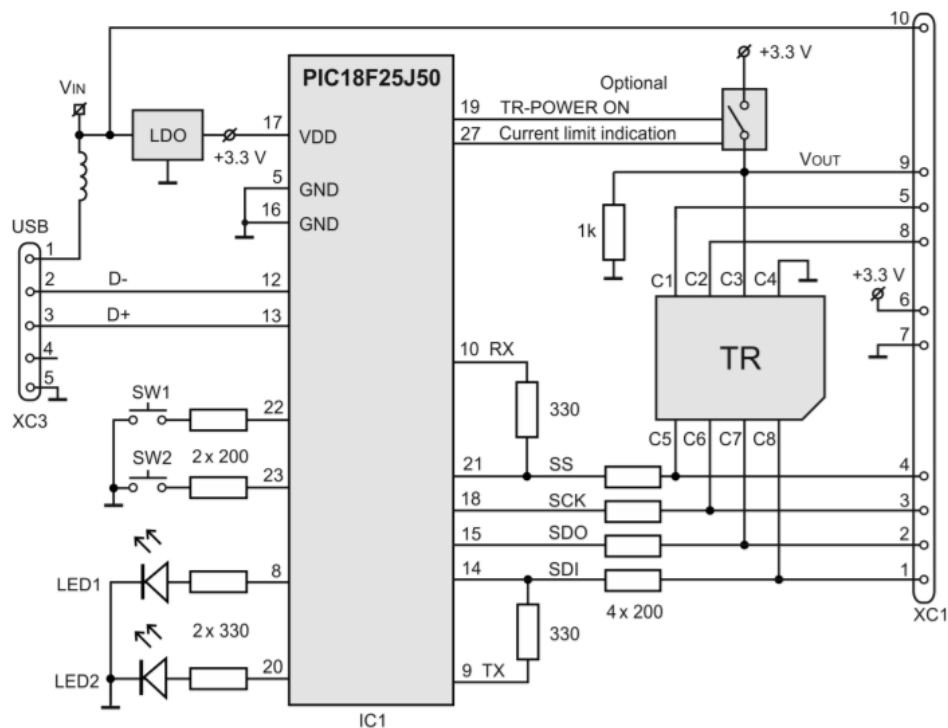


Obrázek 2: Vývojový kit DS-START-04A [1]

### 2.1.1 DK-EVAL-04A

DK-EVAL-04A je rozměrově malý, univerzální vývojový kit pro bezdrátové aplikace s transceivery. Slouží k ladění a testování IQMESH sítě. Tento kit je napájen ze zabudovaného LiPol akumulátoru, v případě vybití se akumulátor dobíjí za pomoci micro USB konektoru a interního dobíječe. Součástí kitu je i zabudovaný integrovaný obvod proti přepětí, podpětí akumulátoru a stabilizátor napětí. Pro připojení transceiverů se využívá 8pinový SIM konektor, dále obsahuje 10pinový konektor s osmi I/O a výstupy pro napájení nízkopříkonových periférií. Důležitou součástí jsou 2 tlačítka pro restartování a připojení uzlů. Indikace restartování, připojení a dobíjení je signalizována za pomoci 3 červených LED.





Obrázek 4: Zjednodušené schéma CK-USB-04A

Uzly jsou do sítě přidávány tzv. bondováním za pomoci tlačítka, kde každému uzlu je přiřazena unikátní adresa a následně se vytvoří cesty k dalším uzlům (routování) za pomoci funkce *Discovery*. Každý uzel může být součástí jiné sítě (maximálně 2 sítí). Pakety jsou přístupné pro adresované uzly v dané síti. Použití DPA je možná pouze v případě síťového módu. Výhodou IQMESH sítě je dosah, robustnost, spolehlivost a odstraňuje potenciální problémy hvězdicové topologie.

## 2.3 Způsoby programování

Transceivery se programují za pomoci hardwarového programátoru CK-USB-04A. Transceiver se zasune do SIM konektoru programátoru, ten však nesmí být připojen k napájení, protože by mohlo dojít ke zničení modulu, a poté k USB sběrnici.

Pro naprogramování transceiverů může být využito dvou způsobů, buď řízený datovým tokem s využitím DPA (Custom DPA handler) a HWP, nebo řízený uživatelskou aplikací. [1]

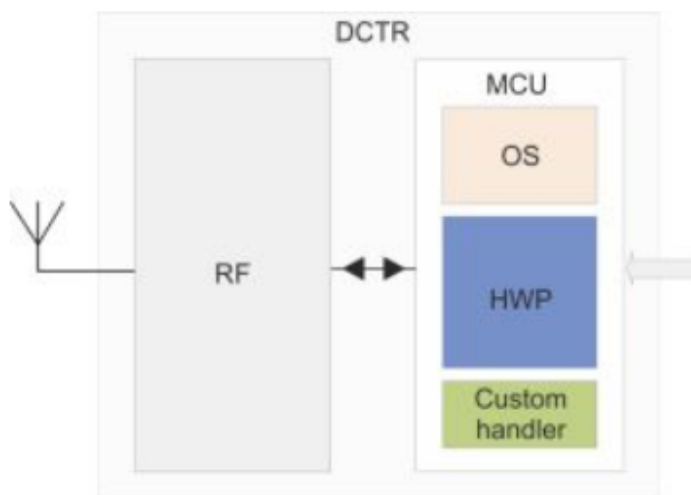
### 2.3.1 DPA

DPA je jednoduchý bytově orientovaný protokol používaný pro řízení periférií, u kterého není nutný zásah programátora. Pomocí příkazů jsou prováděny vybrané operace, čímž se tato metoda stává rychlejší a snadnější. Koordinátor pošle požadavek pro změření dané veličiny, pokud nedošlo k výpadku, obdrží potvrzovací zprávu s naměřenými daty. Rychlost odpovědi je závislá na velikosti sítě a typu hardwarového profilu. U LP HW je odpověď časově náročnější.

DPA patří do třívrstvé architektury, která se skládá z OS, DPA vrstvy a Custom DPA handleru. Custom DPA handler slouží k rozšíření funkcionality HWP a programuje se pomocí jazyka C. V případě použití musí být Custom DPA handler povolen v konfiguraci transceiveru. Aplikace napsaná za pomoci Custom DPA handleru může např. posílat příkazy a přijímat odpovědi a data přes UART rozhraní. DPA metoda podporuje IQMESH síť. [1]

DPA paket se skládá z pěti hlavních částí:

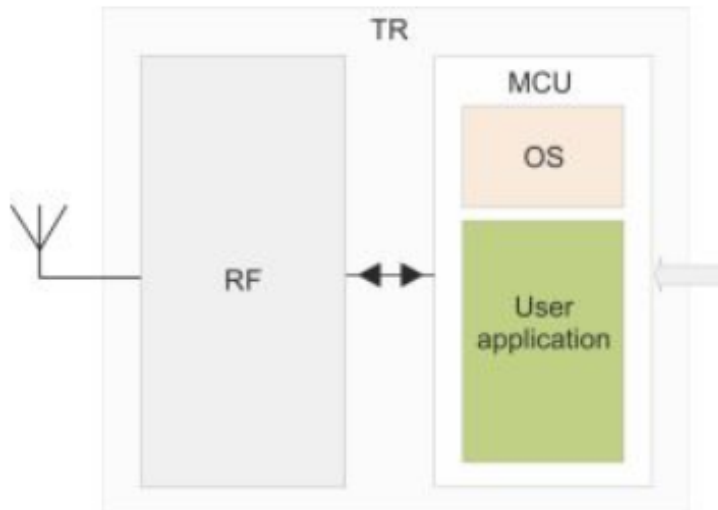
- NADR – adresa síťového zařízení (velikost 2 B)
- PNUM – číslo periférie (velikost 1 B)
- PCMD – příkaz určující akci, která má být provedena (velikost 1 B)
- HWPID – ID HW profilu, jednoznačně určuje funkčnost zařízení, akci provede pouze zařízení se stejným HWPID jako DPA požadavek (velikost 2 B)
- PData – uživatelská data, pole bytů (maximálně 56 B)



Obrázek 5: Způsob programování - využití DPA

### 2.3.2 OS

Dvouvrstvá architektura, která obsahuje OS a uživatelskou aplikaci napsanou v jazyce C. Nepodporuje IQMESH síť, je více flexibilní, ale vyžaduje větší znalosti a schopnosti programátora.



Obrázek 6: Způsob programování - přístup přes OS

## 2.4 Softwarové prostředky

Pro práci s IQRF transceivery je nutnost použití vývojového prostředí IQRF IDE, které slouží k nahrávání hardwarových profilů, firmwaru a DPA konfigurace. Pro editaci zdrojového kódu lze využít libovolného editoru.

### 2.4.1 IQRF IDE

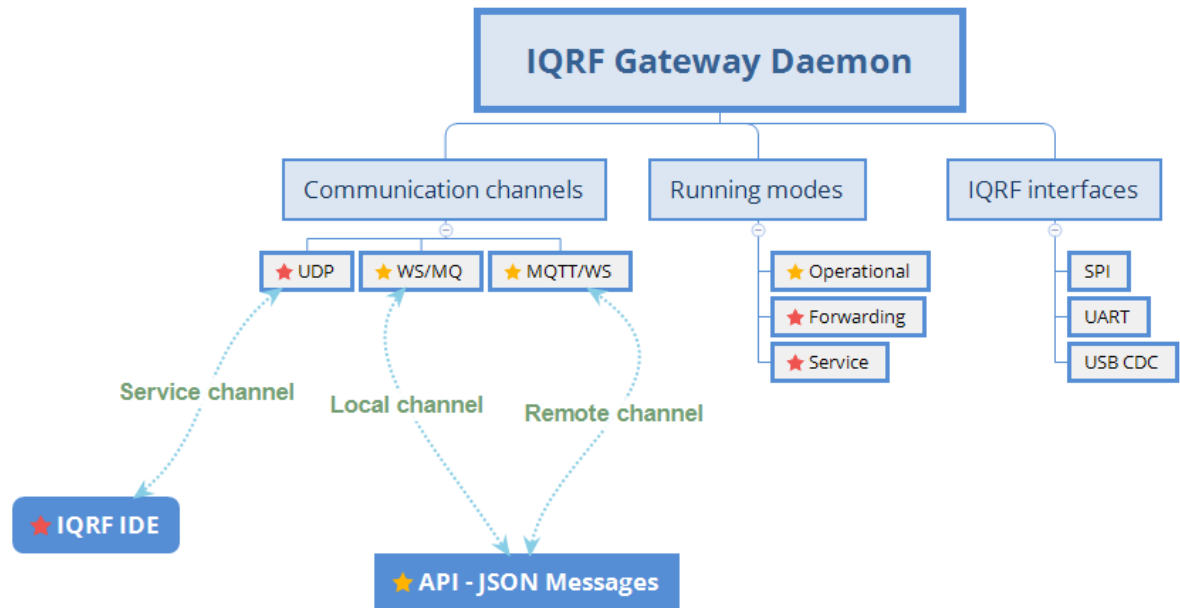
Projektově orientované vývojové prostředí vyvinuté firmou IQRF, určené pro vývojáře vyžadující komplexnost, interaktivní a sofistikované nástroje. Obsahuje nástroje pro návrh, programování, ladění, údržbu, upgrade OS a firmware IQRF transceiverů, dále RF komunikační terminál, správce IQMESH sítě, terminálový log a správce paketů. Všechny IQRF vývojové kity a brány s USB rozhraním jsou podporovány. Nepodporuje transceivery řady TR-5xB a nižší. Součástí instalace je i kompilátor pro jazyk C. Další možností je použití příkazového řádku, která je pro zkušenější vývojáře rychlejší a efektivnější.

### 2.4.2 Visual Studio Code

Jedná se o jednoduchý multiplatformní editor zdrojových kódů vyvinutý firmou Microsoft, je poskytován zdarma a jako open source i pro komerční použití. Nabízí mnoho nástrojů na ladění programů, zvýrazňování syntaxe a nabízí možnost doinstalování snippetů pro usnadnění psaní kódu. Pro editaci firmwaru měřicího uzlu byl dostačující.

## 2.5 IQRF GW Daemon

IQRF GW Daemon je open source softwarový balíček umožňující jednoduše vytvořit IQRF bránu s internetovým připojením pomocí zařízení s OS Linux. Obvykle je provozován na vestavěném jednočipovém počítači (Raspberry Pi, UpBoard, Linux PlugPC, BeagleBone). Může být s transceiverem spojen pomocí SPI, UART nebo USB CDC. Na transceiveru je implementován hardwarový profil koordinátora IQMESH sítě. IQRF GW Daemon využívá několik komunikačních kanálů (UDP, WebSocket, MQTT) a může být nastaven pomocí IQRF Gateway Webapp.



Obrázek 7: Přehled funkcionalit IQRF GW Daemona

### 3 Platforma Raspberry Pi

Raspberry Pi je jednodeskový počítač o velikosti přibližně platební karty vyvinutý britskou nadací Raspberry Pi Foundation v roce 2012. Důvodem vzniku bylo podpořit výuku informatiky a seznámit studenty s různými možnostmi řízení zařízení. Existuje několik modelů Raspberry Pi, nejnovější je třetí generace model B+ z roku 2018.

V bakalářské práci byl využit Raspberry Pi 3 model B, který je osazen 64bitovým čtyřjádrovým procesorem ARM Cortex-A53 1,2 GHz, operační paměť má velikost 1 GB a je sdílená s grafickým procesorem. Výkonnostně je srovnatelný se slabším stolním počítačem. Obsahuje HDMI konektor pro připojení monitoru, čtyři USB porty, jeden ethernetový a komponentní port, čtyřicet pinů pro připojení různých periférií, port pro připojení kamery a displeje, integrovaný Wi-Fi a Bluetooth modul. Primárně je navržen pro různé distribuce OS Linux, operační systém je zaváděn z microSD karty a zařízení je napájeno pomocí externího zdroje přes mikro USB konektor. Cena tohoto počítače se pohybuje kolem 1000 Kč.



Obrázek 8: Platforma Raspberry Pi 3 model B



## 4 Snímač prašnosti Honeywell HPM115S0

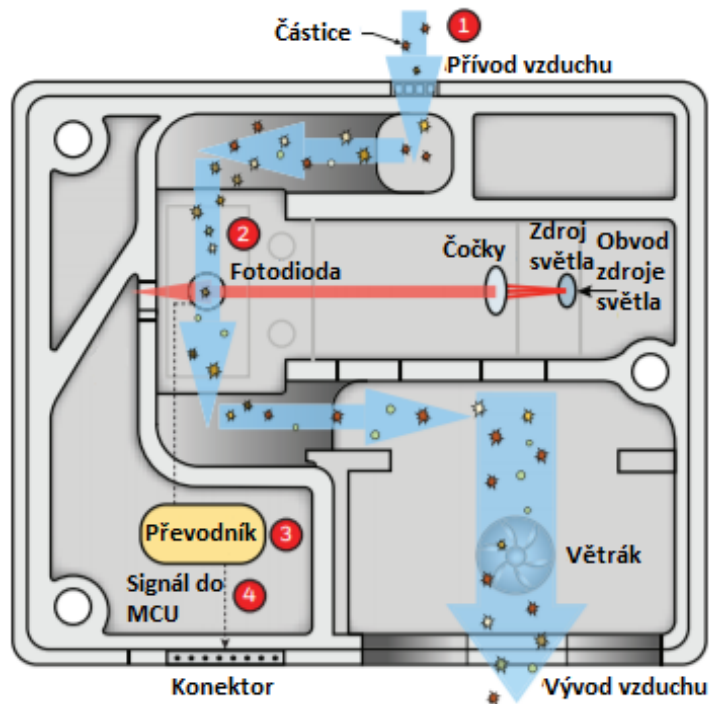
Průmyslový snímač prašnosti HPM115S0 slouží k monitorování znečištění prostředí prachem a jelikož splňuje požadavky EMC, je vhodný pro měření prašnosti v téměř jakémkoliv prostředí. Vyznačuje se rychlou odezvou v reálném čase, odolností, dlouhou životností, splňuje EMC požadavky pro těžký průmysl a je v souladu se směrnicemi RoHS a REACH Evropského parlamentu a Rady. Ke svému provozu vyžaduje napájecí napětí 5 V a maximální spotřeba proudu se pohybuje do 80 mA v režimu měření. Komunikace přes rozhraní UART je založena na 3,3V logice. [3]



Obrázek 9: Snímač prašnosti Honeywell HPM115S0

### 4.1 Princip

Jedná se o laserový snímač, který využívá principu rozptylu světla k detekci množství částic v koncentraci od  $0 \mu\text{g}/\text{m}^3$  do  $1000 \mu\text{g}/\text{m}^3$  v daném prostředí. Ventilátor nasává vzduch do vstupní části, kde skrze paprsek světla proudí prachové částice. V osvětlovací komoře laserový paprsek osvítí proudící částice, tyto částice částečně paprsek zastíní a fotocitlivý snímač detekuje, jak moc byl paprsek částicemi zastíněn (utlumen). Signál z fotocitlivého snímače je následně poslán do převodníku, kde je zpracován vnitřní elektronikou, která vyhodnotí hustotu znečištění. Interní mikroprocesor a lokální algoritmus naměřená data převedou do strukturovaného datového protokolu. Tento datový protokol je poté odeslán přes rozhraní UART. Snímač je schopen detekovat částice menší jak  $2,5 \mu\text{m}$ . Z koncentrace částic do průměru  $2,5 \mu\text{m}$  dokáže vypočítat množství částic do průměru  $10 \mu\text{m}$ . [3]



Obrázek 10: Struktura snímače

## 4.2 Komunikace - UART

UART (USART) je sériové rozhraní, které slouží ke komunikaci mezi zařízeními a může být nastaveno do dvou režimů. Prvním režimem je asynchronní režim, který se používá pro linky RS232 a RS485, druhým režimem je režim synchronní. Tento typ sériového rozhraní se nachází téměř ve všech mikrokontrolérech. Obvyklá hodnota logické úrovně bývá 5 V nebo 3,3 V.

Asynchronní přenos vysílá data na pinu označovaném jako TX a přijímá data na pinu RX. Pokud nejsou data vysílána ani přijímána, jsou piny nastaveny na hodnotu logické úrovně 1. Komunikace se zahajuje změnou signálu na hodnotu logické úrovně 0, což se označuje jako startovací bit, následuje datový bit s nejnižší vahou po bit s nejvyšší vahou a za bitem s nejvyšší vahou následuje tzv. stop bit, který má vždy hodnotu logické úrovně 1. Po tomto přenosu může začít přenos dalšího bajtu. V asynchronním módu je UART nakonfigurován jako plně duplexní komunikace a v tomto módu může komunikovat s periferiemi (např. osobní počítač). V tomto módu lze data přijímat a zároveň vysílat.

V synchronním módu je USART nakonfigurován jako polo duplexní komunikace, která může komunikovat s periferiemi (např. A/D převodník), navíc lze zařízení nastavit jako master nebo slave, dále používá ke komunikaci datovou a hodinovou linku. Datová linka slouží k přenosu dat a hodinová linka určuje přenos. V tomto módu nelze data přijímat a zároveň vysílat. [2]

Snímač prašnosti využívá pro přenos dat asynchronní režim s rychlostí 9600 baudů, 1 bajt obsahuje 8 databitů a jeden stop bit, paritní bit není v protokolu použit. Snímač zahajuje měření po přijetí konkrétního příkazu. Příkazy jsou ve většině případů čtyřbajtové a hodnota logické úrovně je 3,3 V.

Tabulka 1: Struktura příkazu pro čtení ze snímače [3]

Čtení	Head	Len	Cmd	Data	CS
Příkaz	0x68	0x01	0x04	NA	$CS = \text{MOD}((65536 - (\text{HEAD} + \text{LEN} + \text{CMD} + \text{DATA})), 256)$

#### 4.2.1 Struktura datového protokolu

Protokol využitý v mikroprocesoru snímače reprezentuje naměřená data ve specifickém datovém formátu o délce 32 bajtů. Nulový a první bajt obsahují hlavičku, druhý a třetí bajt obsahuje informace o délce rámce, šestý až devátý bajt obsahuje informace o naměřené koncentraci částic v ovzduší. Poslední dva bajty slouží ke kontrolnímu součtu, ostatní bajty jsou prázdné. Z těchto posledních dvou bajtů lze programově zjistit, zda byl přenesen celý rámec beze ztráty nějakého bajtu. [2]

Před odesláním naměřených dat je nutné zahájit měření, poté přečíst naměřená data, a nakonec ukončit měření. Tohoto lze dosáhnout za pomoci speciálních příkazů. Na každý příkaz snímač odpoví pozitivní (0xA5A5) nebo negativní (0x9696) odpovědí. Výjimkou je pozitivní odpověď na čtení, kdy snímač posílá naměřená data.

Každý příkaz obsahuje:

- HEAD – hlavička, určující začátek dat. rámce
- LEN – výchozí hodnota 0x01, při práci s koeficientem přizpůsobení zákazníka 0x02
- CMD – příkaz ve formě hexadecimálního čísla, různé pro každý příkaz
- Data
- CS – kontrolní součet

Po přečtení datového protokolu je nutné naměřená data přepočítat z hexadecimálních hodnot na hodnoty dekadické. Pro získání reálné koncentrace částic do 2,5 mikronů je nutné vynásobit obsah šestého bajtu hodnotou 256 a přičíst hodnotu sedmého bajtu. U výpočtu reálné koncentrace částic do 10 mikronů je užito stejného postupu pro osmý a devátý bajt.

Následující tabulka zobrazuje výstupní datový protokol, který snímač odesílá po přijetí příkazu pro čtení naměřených hodnot.

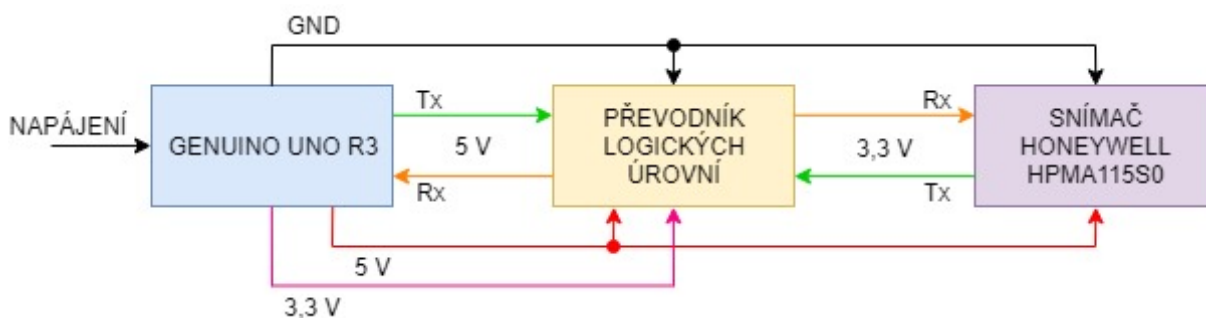
Tabulka 2: Struktura datového protokolu [3]

Byte number	Head0	Head0	Head0
Byte0 Byte1	Head0 Head1	0x42 0x4d	fixed
Byte2 Byte3	Len_H Len_L	... ...	Frame length=2x13+2(data length+CS length)
Byte4 Byte5	Data0_H Data0_L	... ...	reserve
Byte6 Byte7	Data1_H Data1_L	... ...	PM2.5 concentration
Byte8 Byte9	Data2_H Data2_L	... ...	PM10 concentration
Byte10 Byte11	Data3_H Data3_L	... ...	reserve
Byte12 Byte13	Data4_H Data4_L	... ...	reserve
Byte14 Byte15	Data5_H Data5_L	... ...	reserve
Byte16 Byte17	Data6_H Data6_L	... ...	reserve
Byte18 Byte19	Data7_H Data7_L	... ...	reserve
Byte20 Byte21	Data8_H Data8_L	... ...	reserve
Byte22 Byte23	Data9_H Data9_L	... ...	reserve
Byte24 Byte25	Data10_H Data10_L	... ...	reserve
Byte26 Byte27	Data11_H Data11_L	... ...	reserve
Byte28 Byte29	Data12_H Data12_L	... ...	reserve
Byte30 Byte31	Checksum_H Checksum_L	... ...	CS=Head0+Head1+Len_H+...+Data12_L

## 5 Testování

Za účelem otestování funkčnosti a komunikace snímače prašnosti bylo využito klonu Arduino Uno připojeného k osobnímu počítači přes USB port. Snímač byl k Arduino připojen přes převodník logických úrovní, protože jeho hodnota logické úrovně pro komunikaci byla navržena pro 3,3voltovou logiku, kdežto Arduino pracuje s 5voltovou logikou. Jelikož toto Arduino má pouze jednu sériovou linku, která byla využita pro komunikaci s osobním počítačem, musela být vytvořena tzv. softwarová sériová linka.

Vývojová deska Arduino Uno je postavena na mikrokontroléru ATmega328P, obsahuje 14 digitálních vstupně/výstupních pinů a 6 analogových vstupů. Mikrokontrolér je taktován na 16 MHz. Deska může být napájena z externího zdroje nebo za pomoci USB konektoru.

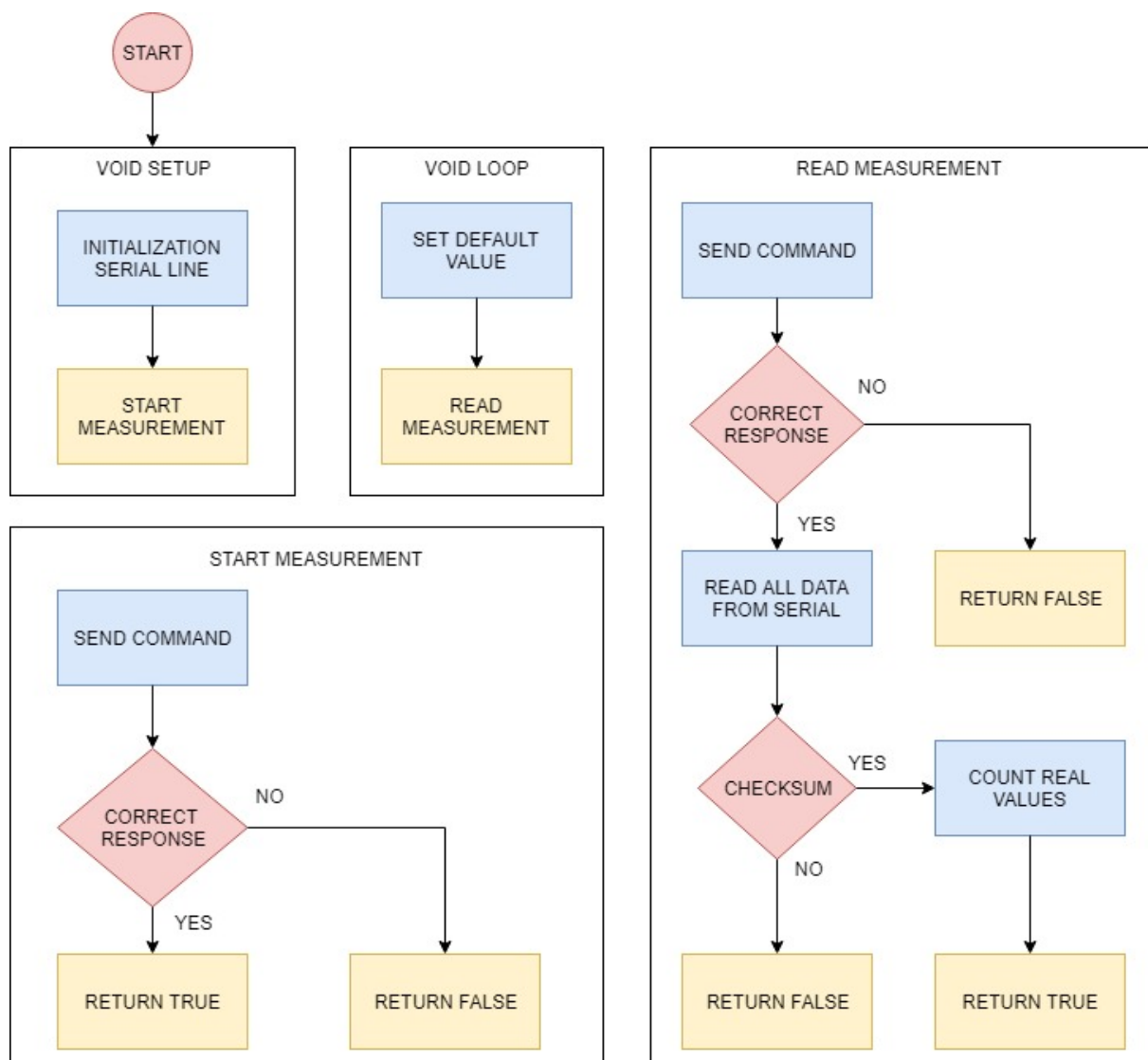


Obrázek 11: Blokové schéma zapojení

Softwarová sériová linka byla vytvořena pomocí knihovny SoftwareSerial, která dokáže udělat další kopii sériové linky na jiných digitálních pinech. Užití této metody není zcela vhodné, protože při použití více sériových linek u mikrokontroléru s jednou sériovou linkou můžeme v daném okamžiku pracovat pouze s jednou jedinou sériovou linkou, pro testování však byla dostačující. Při použití softwarově vytvořené sériové linky není zaručeno, že data budou přenesena bez chyby. Může totiž nastat situace, kdy prioritní sériová linka přeruší čtení ze softwarové a bude přenesena jen část dat. Z tohoto důvodu bylo v programu využito kontrolního součtu, kde se porovnával kontrolní součet poslaný ze snímače a vypočtený kontrolní součet v programu Arduina podle vzorce od výrobce snímače. [8]

Program začíná přilinkováním knihoven, nastavením pinů pro softwarovou sériovou linku a vytvořením globálních proměnných pro uložení informace o znečištění a indikaci chyby. V inicializační části programu, která proběhne pouze jednou po spuštění, se inicializuje sériová linka pro komunikaci s PC a softwarová sériová linka s parametry pro rychlost komunikace. Dodržení rychlosti komunikace dané výrobcem snímače je důležité, v případě nastavení nesprávné rychlosti nebude komunikace správně fungovat. V nekonečné smyčce programu se v každém cyklu nastavují hodnoty proměnných pro uložení naměřených dat na výchozí hodnotu, dále se zde "vyvolává" funkce, která zajišťuje měření. Pokud jsou přečtená data v pořádku, vypíše se do konzole v počítači hláška s informací o úspěšném přečtení dat a tato data jsou následně vypsána.

V testovacím programu probíhalo měření každých 5 sekund a celé zařízení bylo otestováno na nepájivém kontaktním poli.



Obrázek 12: Vývojový diagram softwaru pro testování na vývojové desce Arduino

Funkce pro zahájení měření posílá přes softwarově vytvořenou sériovou linku pole bajtů, které obsahuje příkaz, díky kterého se snímač připravuje na měření. Pokud snímač obdrží zprávu bez chyby a podaří se mu zahájit měření, pošle přes sériovou linku 2 bajty, které definují, zda vše proběhlo v pořádku či nikoliv.

Vyčítání dat ze snímače bylo rovněž realizováno za pomoci funkce, která posílala pole bajtů pro určení příkazu. Po odvysílání na sériovou linku program okamžitě začal číst odpověď a pokud nenarazil na specifickou hlavičku dat, která je definovaná v datasheetu výrobce, okamžitě funkci "opouštěl" a vyhodnotil měření jako chybné. Pokud program narazil na korektní strukturu hlavičky a přečetl i zbytek dat ze sériové linky, tak provedl kontrolní součet. Logika kontrolního součtu spočívá v odečtení hodnot jednotlivých bajtů od čísla 65 536 kromě posledních dvou bajtů. Pokud byl zbytek po celočíselném dělení roven hodnotě posledních dvou bajtů, kontrolní součet byl v pořádku a proběhl přepoččet jednotlivých koncentrací na reálné hodnoty.

Testovací program byl napsán v jazyce Wiring a vývojovém prostředí Arduino IDE. Zprovoznění snímače na vývojové desce Arduino nebyla nijak zvlášť složitá, protože ke snímači byla napsána knihovna, kterou stačilo vhodně implementovat. Tato knihovna však nemohla být použita při práci s technologií IQRf, protože byla napsána v jazyce C++ a vývojové prostředí IQRf IDE obsahuje kompilátor pouze pro jazyk C. Ve finální verzi firmwaru, která bude následovat v další kapitole, sloužila knihovna jako inspirativní předloha.

```
Read measurement status is 1
PM2.5 value is 66ug/m3
PM10 value is 68ug/m3

Read measurement status is 1
PM2.5 value is 58ug/m3
PM10 value is 60ug/m3

Read measurement status is 1
PM2.5 value is 64ug/m3
PM10 value is 66ug/m3

Read measurement status is 1
PM2.5 value is 69ug/m3
PM10 value is 71ug/m3

Read measurement status is 1
PM2.5 value is 67ug/m3
PM10 value is 69ug/m3

Read measurement status is 1
PM2.5 value is 66ug/m3
PM10 value is 68ug/m3
```

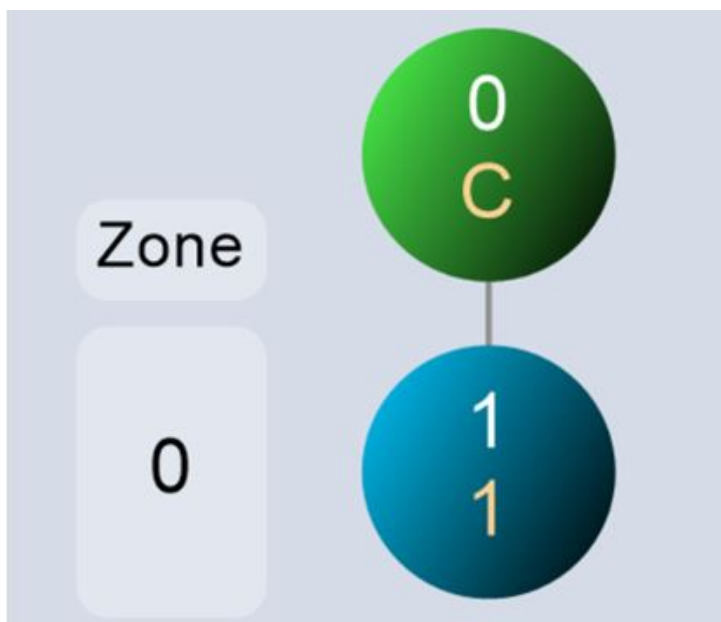
Obrázek 13: Výpis naměřených dat do terminálu

## 6 Návrh a realizace

Cílem bylo vytvořit měřicí a řídicí uzel pro měření znečištění ovzduší, kde by komunikace a přenos dat probíhal bezdrátově s využitím technologie IQRF. Řídicí uzel posílá příkazy měřicímu uzlu a ten posílá naměřená data zpět k řídicímu uzlu. Řídicí uzel následně data vizualizuje.

### 6.1 Měřicí uzel

Je realizován vlastním návrhem DPS, vyměnitelným IQRF transceiverem a step-up měničem, snímačem prašnosti od firmy Honeywell, LiPol baterií, SMA anténou a vše je umístěno v krabici vlastního návrhu. Baterie slouží jako záložní zdroj elektrické energie, transceiver je řídicí částí tohoto uzlu a je k němu připojen snímač přes sériové rozhraní UART, který je napájen step-up měničem. Podrobnosti k jednotlivým součástem byly zmíněny v teoretické části této práce.



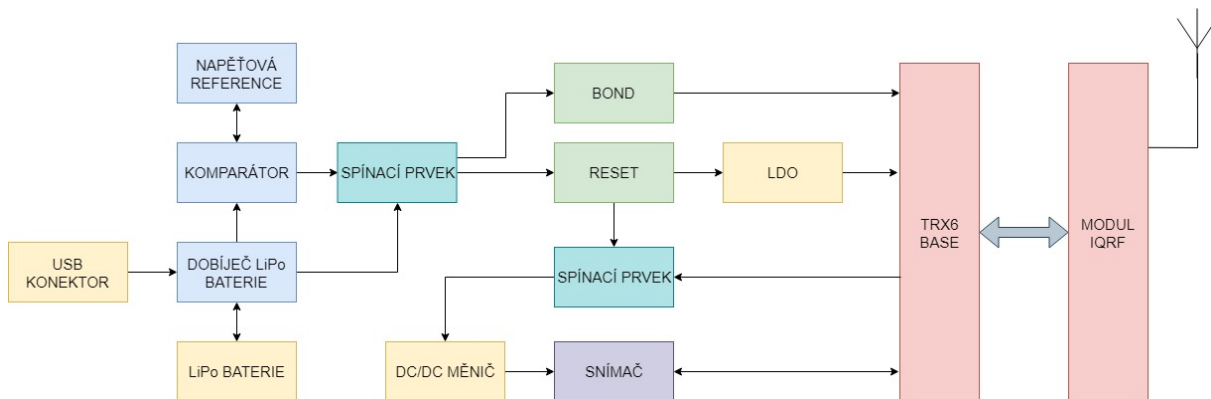
Obrázek 14: Schéma IQMESH sítě

IQRF MESH síť je v tomto případě jednoduchá, protože obsahuje pouze jednoho koordinátora sítě a jeden měřicí uzel. Rozšíření sítě o další měřicí uzly je možné až do maximální kapacity IQMESH sítě. Měřicí uzel obsahuje modul TR-76D a v tomto modulu je nahrán hardwarový profil DPA-Node-LP-7xD-V400-190120 a vlastní firmware jehož základem je CustomDpaHandler-UART. V DPA konfiguraci bylo zvoleno ISM pásmo 868 MHz, komunikační kanál 52, dále typ sítě STD+LP, povolena periferie UART, povolení použití Custom DPA Handleru, Autoexec a vysílací výkon byl zvolen na nejvyšší možnou hodnotu 7. Zabezpečení sítě a šifrování dat nebylo v práci řešeno. Verze operačního systému byla 4.03 a verze DPA 4.xx.



### 6.1.1 Schéma

Návrh elektrického schématu byl inspirován schématem vývojového kitu DK-EVAL-04A. Schéma se skládá z USB mikro konektoru, který slouží k napájení obvodu pro dobíjení LiPol baterie, napětovou referenci 2,5 V a komparátor, který porovnává napětí za dobíječem s napětovou referencí. Hodnota napětí za integrovaným obvodem pro dobíjení LiPol baterie je 3,7 V. Pokud je napětí baterie menší jak referenční napětí, je celé zařízení odpojeno od napájení. Dobíjení je signalizováno za pomoci LED. Dále se zde nachází tlačítka pro resetování zařízení a pro připojení zařízení do MESH sítě (bondovací tlačítko). Proces resetování a bondování je opět signalizován pomocí LED. Jelikož modul TR-76D vyžaduje napájení 3 - 3,4 V, musel být použit LDO (lineární regulátor stejnosměrného napětí, který je napájen napětím blízkým výstupnímu stabilizovanému napětí), který reguluje napětí pro IQRF modul na 3,3 V. Posledním důležitým prvkem je step-up měnič, který zvyšuje napájecí napětí na 5 V pro napájení snímače prašnosti. Tento měnič je dvoustavově řízen modulem TR-76D a samotným firmwarem. Elektrické schéma bylo vytvořeno v programu Eagle 9.2.2 Education.



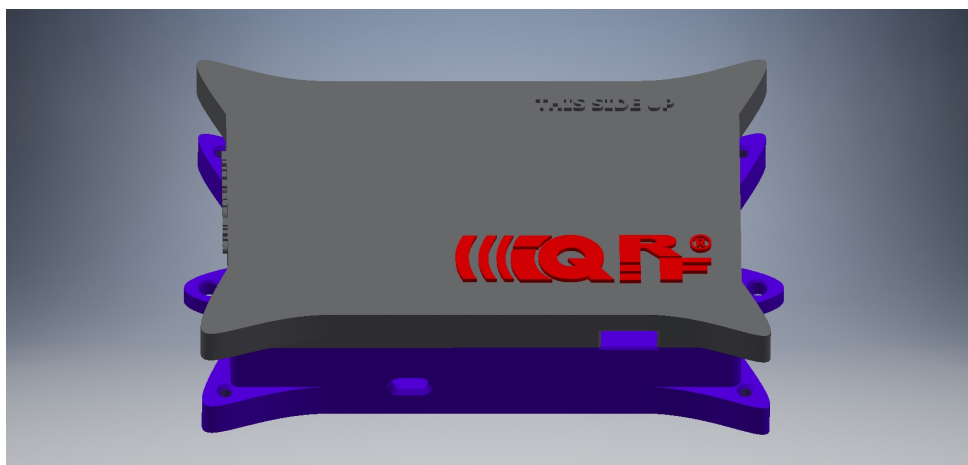
Obrázek 15: Zjednodušené blokové elektrické schéma

### 6.1.2 DPS

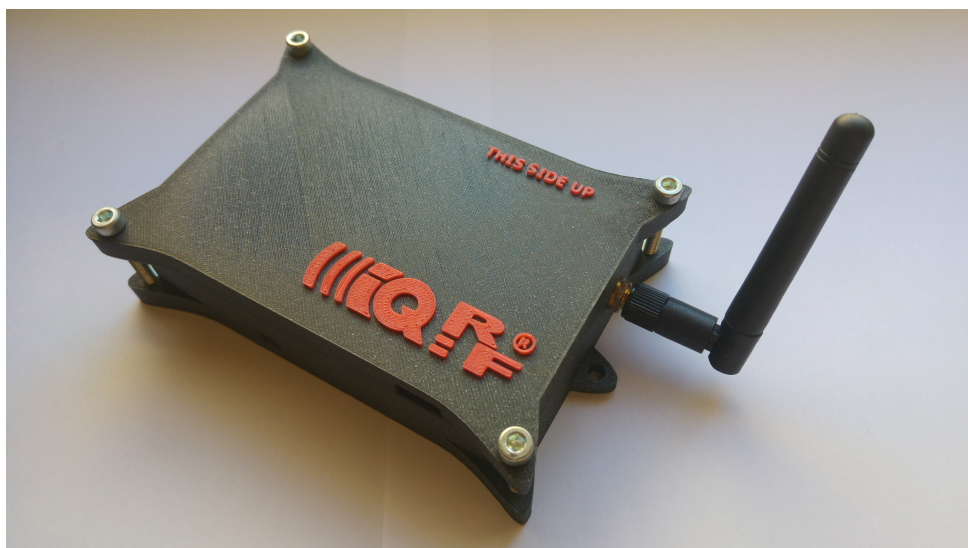
Deska plošných spojů byla navržena pro SMT součástky velikosti 0603 a byla nakreslena ve vrchní vrstvě. Velikost desky je 65x53 mm. Šířka vodivých cest od napájení je 0,4 mm a šířka signálových cest byla zvolena na 0,254 mm. Odpojení obvodu od napájení je vyřešeno pomocí jumperu. Měnič napětí, baterie a modul TR-76D se snímačem prašnosti jsou k desce připojeny pomocí patič a konektorů. Patice byly využity z důvodu jednoduché výměny při poškození jednotlivých částí. Maximální proudový odběr se pohybuje do 100 mA. Maximální protékající proud se dá omezit pomocí rezistoru R4, který je připojen k obvodu pro dobíjení. Osazování desky plošných spojů SMT součástkami probíhalo ruční metodou za pomoci pájecí stanice a horkovzdušné páječky. Pájka byla použita bezolovnatá cínová s příměsí mědi (Sn97Cu3) a tavidlo Flux. Návrh DPS byl realizován za pomoci programu Eagle 9.2.2 Education.

### 6.1.3 Krabička

Krabička byla vymodelována v CAD softwaru Autodesk Inventor 2019 a pro převedení formátu OBJ na GCODE a finální úpravu krabičky bylo využito programu Slic3r PE a PrusaControl. Velikost a celkový vzhled krabičky byl uzpůsoben velikosti desky plošných spojů, snímače, baterie a byl kladen důraz na to, aby byl snímač, DPS a baterie od sebe navzájem odděleny přepážkou. Krabička obsahuje distanční sloupky, aby nebyla DPS položena na dně a dala se dobře přišroubovat. Pro aretaci snímače byly do krabičky umístěny tři aretační kolíky, do kterých se snímač zasune, aby nebylo možné s ním jakkoliv pohybovat. Pro snadné připojení USB kabelu byla navržena díra tak, aby tiskárna neměla problémy s tiskem. Díry pro přívod a odvod vzduchu byly rovněž navrženy s ohledem na snadný a bezproblémový tisk.



Obrázek 16: Návrh krabičky v Autodesk Inventor 2019



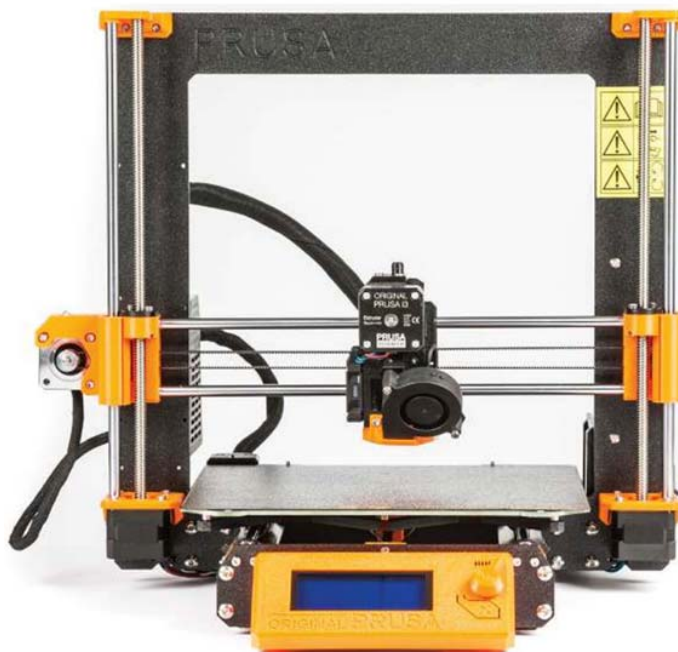
Obrázek 17: Vytištěná krabička

Tisk krabičky byl realizován 3D tiskárnou Prusa i3 mk3, která byla sestavena ze stavebnice a zkalibrována na testovacích objektech, aby výsledná krabička byla co nejpěknější. Doba tisku při optimálním nastavení detailů na 0,15 mm, hustotě výplně na 100 % byla necelých 7 hodin i s tiskem podpor a límce. Velikost trysky hot-endu byla 0,4 mm. Jako materiál pro tisk krabičky byl zvolen filament typu PLA, který je nejčastěji používaným materiálem pro 3D tisk. Mezi hlavní výhody PLA patří:

- Snadný tisk
- Detailní tisk drobných modelů
- Bezproblémový tisk velkých součástek
- Tvrdost a tuhost
- Šetrnost k životnímu prostředí

Jako každý materiál má samozřejmě i nevýhody:

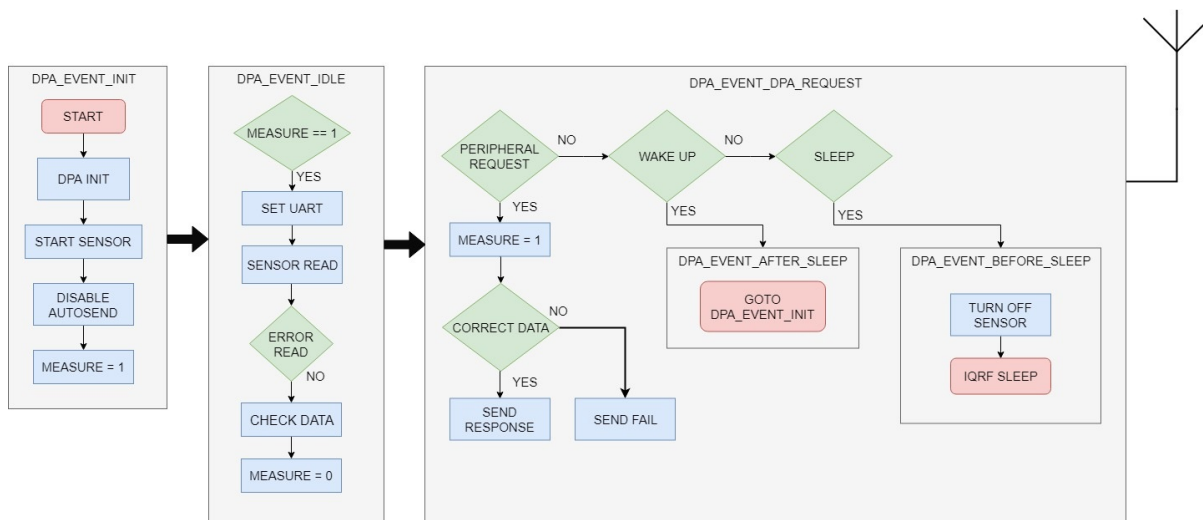
- Křehkost
- Nízká tepelná odolnost
- Složitý post-processing



Obrázek 18: Sestavená tiskárna Prusa i3 mk3

### 6.1.4 Firmware

Řídící částí měřicího uzlu je transceiver TR-76D, který obsahuje osmibitový mikrokontrolér PIC16LF1938-I/MV. Program pro tento mikrokontrolér je napsán v jazyce C za pomoci frameworku DPA. Transceiver byl nejprve naprogramován pro standardní hardwarový profil a poté byl software přizpůsoben pro fungování v LP módu, což je energeticky méně náročný hardwarový profil, ale není možné využívat všech funkcí DPA frameworku. Firmware se skládá ze tří hlavních částí *DpaEvent\_Init*, *DpaEvent\_Idle* a *DpaEvent\_DpaRequest*.



Obrázek 19: Vývojový diagram firmwaru

Základem programu je CustomDpaHandler-UART, což je předpřipravená šablona obsahující makro, které zajišťuje nastavení komunikace přes sériové rozhraní UART. Tímto je zjednodušená následná práce s registry, dalšími parametry komunikace a dále se nemusí nic nastavovat. Dále program obsahuje řadu jednobytových konstant, které se v programu posílají jako příkazy pro snímač. Důvodem tohoto řešení byla nemožnost vytváření polí v DPA, a proto se musely tyto bajty posílat každý zvlášť. Po připojení transceiveru k napájení se nejprve provede blok programu *DpaEvent\_Init*, kde jsou nastaveny vstupní a výstupní piny, dále zde program čeká na spuštění snímače, nastaví sériové rozhraní pro posílání příkazů, pošle příkaz pro zapnutí snímače a zakáže snímači automaticky posílat naměřená data. Pokud vše proběhne v pořádku, povolí se následné čtení dat pro příští blok. Následně program tento blok opouští a pokračuje do bloku *DpaEvent\_Idle*. Blok *DpaEvent\_Init* se v cyklické části programu dále nespouští, dokud nedojde k restartování zařízení. [9]

*DpaEvent\_Idle* zajišťuje vyčítání naměřených dat ze snímače. Pokud předchozí blok povolil následné měření, program pošle snímači příkaz pro čtení naměřených dat a ihned zkontroluje, zda nenastala chyba při zápisu a čtení sériového rozhraní UART. Chyba je signalizována pěti-násobným bliknutím červené LED.

V případě korektního odeslání a přijetí dat jsou přijatá data přečtena do paměti RAM a následně jsou po jednom bajtu přiřazena proměnným, které odpovídají jednotlivým bajtům datového protokolu. Kopírování do paměti RAM je signalizováno bliknutím červené a zelené LED. Tento blok proběhne před tím, než hlavní smyčka obdrží požadavek od řídicího uzlu.

Poslední částí programu je blok *DpaEvent\_DpaRequest*, který probíhá, pokud je přijat požadavek z řídicího uzlu na uživatelskou periférii. Obsahuje ještě jednu kontrolu, zda byla všechna důležitá data přiřazena jednotlivým proměnným. Jestliže kontrola proběhne v pořádku, jsou data přiřazena datové části odpovědi na požadavek a poslána řídicímu uzlu. Jestliže jednotlivé proměnné obsahují nesprávná nebo žádná data, datová část odpovědi obsahuje nulové hodnoty.

Struktura DPA požadavku z řídicího uzlu do měřicího uzlu obsahuje adresu nodu (NADR), číslo definující periférii (PNUM), příkaz pro danou periférii (PCMD) a identifikační číslo hardwarového profilu (HWPID). Odpověď z měřicího uzlu obsahuje adresu nodu (NADR), ze kterého je odpověď poslána, číslo definující periférii (PNUM), příkaz pro danou periférii (PCMD), identifikační číslo hardwarového profilu (HWPID), číslo chyby (ERRN), DPA odpověď (DPA) a data (PDATA). V níže uvedených tabulkách lze vidět, jak vypadá reálná struktura DPA požadavku a odpovědi. Hodnoty NADR, PNUM, PCMD a HWPID se ani v jednom případě nemění.

Tabulka 3: Struktura DPA požadavku z řídicího uzlu - požadavek na uživatelskou periférii

<b>NADR (Lo)</b>	<b>NADR (Hi)</b>	<b>PNUM</b>	<b>PCMD</b>	<b>HWPID (Lo)</b>	<b>HWPID (Hi)</b>
0x01	0x00	0x20	0x00	0xFF	0xFF

Tabulka 4: Struktura DPA odpovědi z měřicího uzlu

<b>NADR (Lo)</b>	<b>NADR (Hi)</b>	<b>PNUM</b>	<b>PCMD</b>	<b>HWPID (Lo)</b>	<b>HWPID (Hi)</b>	<b>ERRN</b>	<b>DPA</b>
0x01	0x00	0x20	0x00	0xFF	0xFF	0x00	0x4D

Tabulka 5: Struktura DPA odpovědi z měřicího uzlu, uživatelská data

<b>PDATA</b>				
<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
0x00	0x73	0x00	0x76	0x30



## 6.2 Řídicí uzel

Hlavní částí řídicího uzlu je jednodeskový počítač Raspberry Pi 3B, na kterém je nainstalován operační systém Raspbian bez grafického rozhraní, kde je spuštěn IQRF Gateway Daemon, který slouží jako brána mezi mesh sítí a ethernetovou sítí. Tento daemon komunikuje s koordinátorem mesh sítě, který je k Raspberry Pi připojen pomocí sériové sběrnice SPI. Dále komunikuje s vizualizační aplikací pomocí komunikačního protokolu MQTT.



Obrázek 20: Realizace řídicího uzlu

### 6.2.1 Koordinátor sítě

Koordinátor sítě je tvořen transceiverem TR-72D, do kterého byl nahrán hardwarový profil DPA-Coordinator-SPI-7xD-V400-190120 bez jakéhokoli zdrojového kódu. Konfigurace DPA je téměř totožná s konfigurací transceiveru v měřicím uzlu, rozdílem je zakázání Custom DPA Handleru, protože žádný neobsahuje. Další nutností je nastavení stejného komunikačního kanálu a přepnutí modulu do módu USB CDC IQRF v prostředí IQRF IDE. Modul je připojen k Raspberry Pi pomocí konektoru KON-RASP-01 od firmy IQRF a komunikuje pomocí sériové sběrnice SPI. Podrobnější nastavování je realizováno ve webové aplikaci IQRF Gateway Daemon Webapp.[5]

### 6.2.2 IQRF Gateway Daemon Webapp

Jedná se o webovou aplikaci napsanou v českém PHP frameworku Nette a slouží pro jednoduchou správu konfigurace brány. Po úspěšné instalaci je dostupná v jakémkoliv internetovém prohlížeči na IP adrese, která byla Raspberry přidělena. Aplikace nabízí možnost vytváření uživatelů s určitými rolemi. Po přihlášení lze nastavovat různá rozhraní, spravovat IQRF síť, vypnout/zapnout služby, nastavit připojení do cloudu apod. Nabízí možnost přibondování nového zařízení pomocí funkce Smart Connect. Díky této možnosti není nutné při procesu připojování zařízení do sítě fyzicky stisknout bondovací tlačítko ani zařízení nijak demontovat. Jediným problémem je nutnost vygenerování unikátního IQRF Smart Connect kódu při programování modulu.

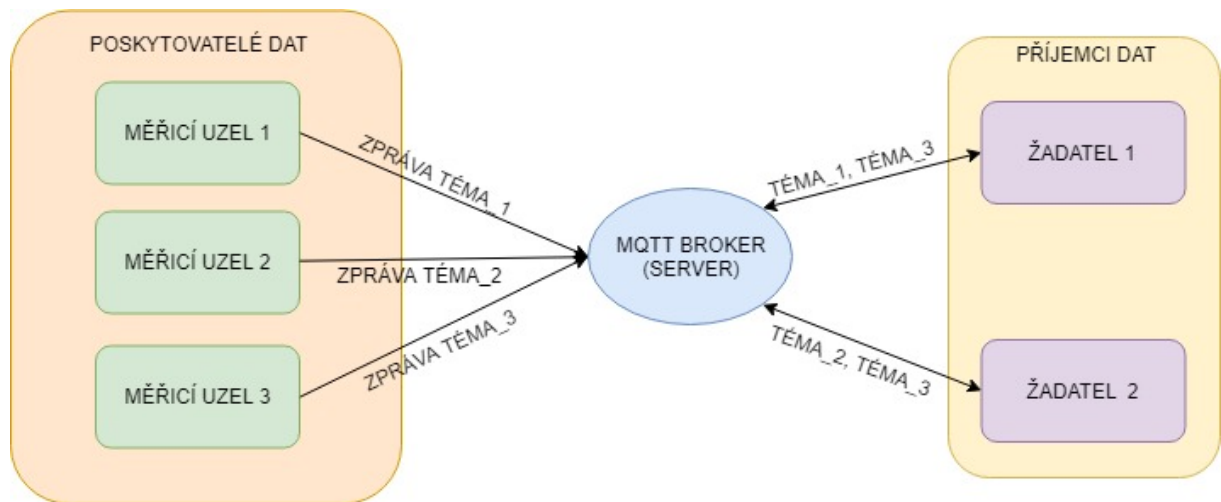
Při vytváření brány bylo důležité mít již předpřipraveného koordinátora sítě připojeného k Raspberry Pi a mít nainstalovaný OS Raspbian Lite. Dalším nezbytným krokem byla aktualizace OS, instalace a povolení přístupu přes službu SSH, povolení SPI rozhraní v konfiguračním souboru operačního systému na Raspberry Pi. Jelikož IQRF GW Daemon a vizualizační aplikace si předávají JSON zprávy, bylo nutné nainstalovat MQTT broker a až poté mohla být provedena instalace IQRF GW Daemonu a webové aplikace. Dále bylo nutné přihlásit se do webové aplikace a vybrat dostupný SPI interface, po kterém bude probíhat komunikace s IQRF modulem.[4]

Posledním krokem byla instalace potřebných součástí pro fungování vizualizační aplikace. Jelikož vizualizační nástroj Node-RED je napsán v jazyce JavaScript na platformě Node.js, musel být do OS Raspberry Pi doinstalován Node.js a Node-RED. Ukázalo se, že nejlepším způsobem je postupovat podle pokynů z oficiálních stránek vizualizačního nástroje Node-RED, protože návod zveřejněný firmou IQRF obsahoval zastaralé balíčky a aplikace následně nefungovala.[4][6]

### 6.2.3 Komunikační protokol MQTT

MQTT je jednoduchý a nenáročný komunikační protokol pro přenos malého objemu dat prostřednictvím běžné TCP/IP internetové sítě a je vhodný zejména pro oblast IoT systémů.

Ve své podstatě funguje na principu předávání zpráv mezi klienty za pomoci centrálního uzlu tzv. brokeru. Může mít několik poskytovatelů dat a několik žadatelů (příjemců), přičemž data posílá pouze těm žadatelům, kteří to vyžadují. Každá zpráva obsahuje tzv. téma (topic), díky kterému jsou data tříděna. Nejčastěji používaným formátem zpráv bývá JSON, použití určitého formátu však není podmínkou a data mohou být libovolného formátu, protože broker s daty nepracuje, pouze je přeposílá, takže je jen na příjemci, aby s daty uměl dále pracovat.[7]



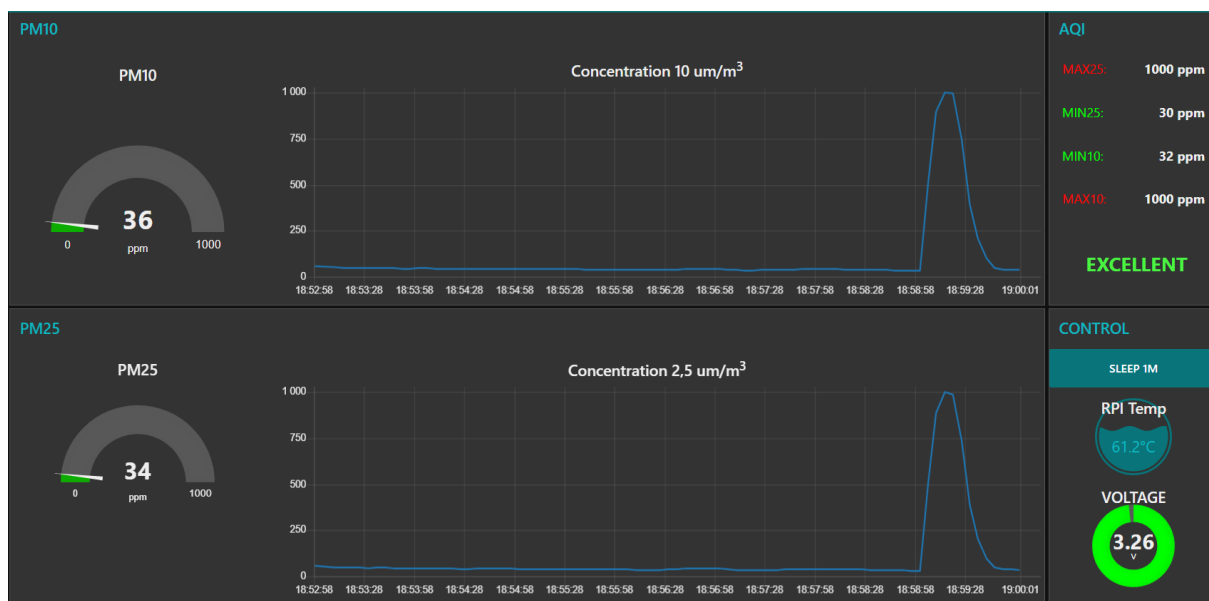
Obrázek 21: Princip fungování MQTT



## 7 Vizualizace

Vizualizační rozhraní bylo vytvořeno v open source programovacím nástroji Node-RED, který pro programování jednotlivých částí využívá flow-based přístupu. Aplikace je složena z uzlů (nodů), kde každý uzel má svůj specifický účel. Na vstup uzlu jsou přivedena vstupní data, uzel tyto data zpracuje a výstupem je pošle k dalšímu uzlu. Takto vzniká datový tok mezi propojenými uzly.

Použití tohoto nástroje umožňuje jednoduše vytvářet aplikace pouhým přetažením předdefinovaného uzlu ze seznamu do pracovní plochy bez nutnosti psaní zdrojového kódu. Samozřejmě je možná i možnost vytváření vlastních uzlů, bez kterých se většinou nelze obejít. Programování vlastních nodů se realizuje za použití skriptovacího jazyka JavaScript. Již vytvořené nody lze ještě dodatečně stylovat pomocí značkovacího jazyka HTML a kaskádových stylů a tím si lze doladit vizualizaci podle vlastních představ. Programovací část vizualizace je dostupná z jakéhokoliv internetového prohlížeče na localhostu na portu 1880. Vizualizační obrazovku je možno zobrazit rovněž na localhost:1880/ui. Datové toky tzv. flows lze jednoduše přenášet díky možnosti exportu pracovní plochy do JSON souborů a ty následně importovat na jiném zařízení.

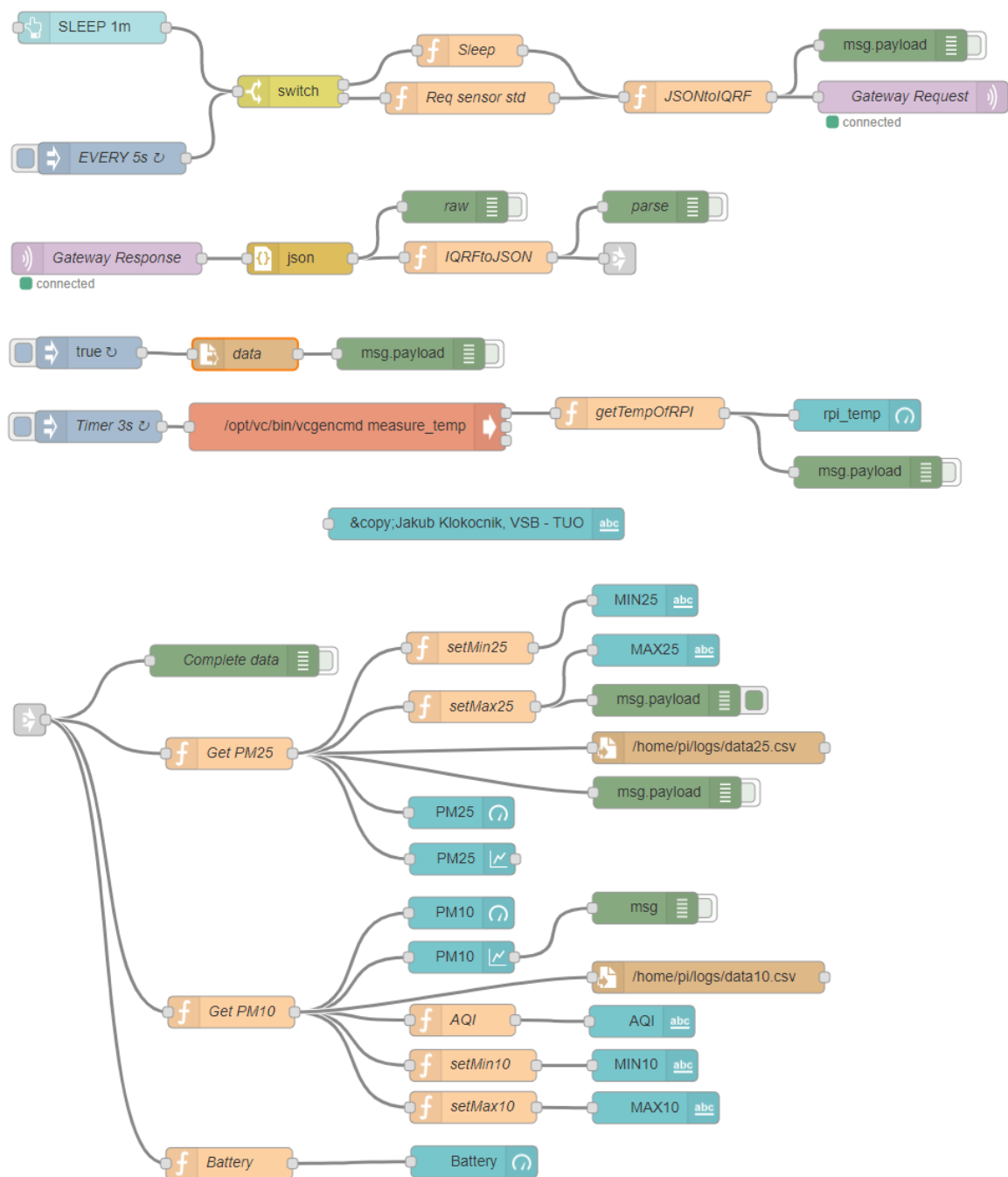


Obrázek 22: Front-end vizualizace

Vstupní část back-endu vizualizace se skládá ze dvou uzlů, kde první uzel je tlačítko s funkcí uspání měřicího uzlu a druhý uzel funguje jako časovač, který v daných okamžicích žádá další uzel, aby poslal požadavek na zahájení měření. Tyto požadavky jsou předány uzlu, který zajišťuje komunikaci mezi řídicím a měřicím uzlem pomocí komunikačního protokolu MQTT. Pokud přijde odpověď od měřicího uzlu, jsou data rozdělena na dílčí části a ty jsou přepočítány a rozparsovány na správný formát.

Hlavními daty jsou zde naměřená koncentrace částic v ovzduší a napětí na baterii měřicího uzlu. Tyto hodnoty jsou po přijetí a zpracování okamžitě vykresleny do grafů. Node-RED si data z grafu dočasně ukládá, ale po restartu Raspberry jsou data z grafů smazána. Množství vykreslených dat záleží na nastavení uzlu v back-endu a je možné jej kdykoliv změnit.

Pro každou koncentraci jsou vytvořeny globální proměnné, do kterých se ukládají naměřená minima, maxima a v každém cyklu jsou porovnávány minulé hodnoty s aktuálními. Naměřené koncentrace znečištění jsou ukládány do souboru CSV, pokud tento soubor neexistuje, je vytvořen nový a uložen na SD kartě na Raspberry Pi. V případě existence tohoto souboru, jsou nově naměřené hodnoty uloženy na konci tohoto souboru. Dále je z množství částic do 10 mikronů vyhodnocována kvalita ovzduší, která je ve vizualizační obrazovce prezentována textovým a barevným zhodnocením kvality ovzduší. Pro kontrolu je ve vizualizaci vykreslována teplota jádra Raspberry Pi. Posledním prvkem je tlačítko, které slouží k přepnutí měřicího uzlu do režimu spánku. Doba usnutí měřicího uzlu je prozatím pevně stanovena na jednu minutu, ale lze ji jednoduchým zásahem v back-endu změnit.



Obrázek 23: Back-end vizualizace

## 8 Zhodnocení výsledků

Realizované zařízení bylo zkompleťováno a namontováno do plastové krabičky, čímž je umožněno ho jednoduše zapojit do provozu popř. připevnit na stěnu díky dvěma montážním otvorům. K plnému nasazení není potřeba do zařízení nijak zasahovat.



Obrázek 24: Měřicí uzel - vnitřní pohled

Měřicí uzel je schopný odpovídat na požadavky každé 3 vteřiny, což v případě měření prašnosti je naprosto dostačující, protože v praxi se toto měření obvykle provádí jednou za hodinu. Tato časová odezva je dána vlastnostmi snímače, který není schopen v kratším intervalu správně reagovat. V případě měření každých 5 vteřin byla úspěšnost měření přes 90 %.

Rozšířením stávajícího řešení by mohlo být vhodnější ukládání dat z vizualizace. Momentálně jsou data zálohována pouze na interní paměť RPI, lepším způsobem by mohlo být ukládání naměřených dat např. do relační databáze MySQL nebo na cloudový server. Cloudovská uložiska jsou většinou poskytována za poplatek, kdežto MySQL databázi je možné realizovat lokálně na RPI popř. s jinými verzemi (Banana Pi), které mají možnost připojení externího uložiska (HDD) přes SATA. Dalším možným vylepšením by mohlo být využití obou vrstev DPS a zmenšení rozměrů celého zařízení.

Softwarovým vylepšením, které by bylo vhodné zrealizovat je ošetření chybových stavů. K těmto stavům dochází jen zřídka, ale řídicí program na ně pouze upozorní a vizualizační aplikace si je zaznamená do grafu, což znamená, že v případě výpadku nebo chybného měření jsou do grafu ve vizualizaci vykresleny nulové hodnoty. Tento nežádoucí stav by bylo vhodné nezaznamenávat a spíše dořešit pomocí alarmových hlášek ve vizualizaci.

V případě rozšíření IQMESH sítě na desítky měřicích uzlů by pravděpodobně nastal problém s komunikací, tudíž by se musela síť přepracovat a řídicí program upravit. Dotazování jednotlivých uzlů na měření v případě větší sítě by nemělo smysl, protože doba odezvy by se pravděpodobně mnohonásobně zvýšila, IQRF však nabízí možnost použití FRC, nicméně toto by vyžadovalo přepracování řídicího programu uzlů a vizualizační aplikace.

## 9 Závěr

Cílem bakalářské práce bylo využití bezdrátové technologie IQRF jako možnost pro přenos naměřených hodnot znečištění ovzduší prachem z měřicího uzlu do uzlu řídicího. Pro realizaci měření je použit optický snímač prašnosti Honeywell HPM115S0.

V rámci řešení bakalářské práce je kompletně navrhnut a vytvořen měřicí uzel. Řídicí uzel je implementován na platformě Raspberry Pi a posílá požadavky na měření měřicímu uzlu, měřicí uzel požadavek zpracuje, obstará měření a zpracuje data ze snímače. Tyto data následně odešle zpět řídicímu uzlu, který naměřená data vizualizuje.

Komunikace mezi jednotlivými uzly je realizována síťovou topologií MESH (IQMESH) s využitím frameworku DPA. Komunikace mezi vizualizační aplikací a IQRF GW Daemonem je uskutečněna pomocí komunikačního protokolu MQTT, který přeposílá JSON zprávy.

Vizualizační aplikace je vytvořena pomocí nástroje Node-RED, skriptovacího jazyka JavaScript a výsledná vizualizační obrazovka je upravena pomocí HTML kódu. Vizualizace je přístupná na localhostu na portu 1880. Platforma Raspberry Pi je připojena do lokální ethernetové sítě, tudíž se lze k vizualizační obrazovce připojit z jakéhokoli zařízení připojeného do stejné sítě.

Realizace této práce mi rozšířila mé dosavadní znalosti. Naučil jsem se pracovat s IQMESH sítěmi, DPA frameworkem a vizualizačním nástrojem Node-RED, o kterých jsem vůbec nic nevěděl. Zdokonalil jsem se v návrhu elektroniky, kreslení DPS, návrhu 3D modelů a programování.

## Literatura

- [1] IQRF Technology [online]. [cit. 2018-12-26]. 2018. Dostupné z: <https://iqr.org/technology>
- [2] Basics of UART Communication [online]. [cit. 2018-12-28]. 2017. Dostupné z: <https://www.electronicshub.org/basics-uart-communication/>
- [3] HPM Series Particle Sensor. Datasheet [online]. [cit. 2018-12-27]. 2018. Dostupné z: <https://sensing.honeywell.com/honeywell-sensing-hpm-series-particle-sensors-datasheet-32322550-e-en.pdf>
- [4] IQRF Gateway RPI [online]. [cit. 2019-02-15]. 2019. Dostupné z: <https://gitlab.iqr.org/alliance/iot-starter-kit/blob/master/install/rpi-board/GW-SbS-INSTALL.md>
- [5] IQRF GW Daemon's documentation [online]. [cit. 2019-02-15]. 2019. Dostupné z: <https://docs.iqr.org/iqr-gateway-daemon/index.html>
- [6] Node-RED documentation [online]. [cit. 2019-02-18]. 2019. Dostupné z: <https://nodered.org/docs/>
- [7] IoT MQTT Prakticky [online]. [cit. 2019-02-19]. 2017. Dostupné z: <https://automatizace.hw.cz/iot-mqtt-prakticky-v-automatizaci-1dil-uvod.html>
- [8] VODA, Zbyšek a HW KITCHEN. Průvodce světem Arduina. Druhé. Bučovice: Martin Stríž, Bučovice, 2017. ISBN 978-80-87106-93-8.
- [9] DPA SW and documentation [online]. [cit. 2019-01-28]. 2019. Dostupné z: <https://iqr.org/technology/dpa/dpa-sw-documentation>

## A Seznam příloh

Příloha v IS EDISON. Adresářová struktura příloh:

- *A\_SCHEMA\_DPS* - elektrické schéma a DPS měřicího uzlu
- *B\_IQRF* - firmware měřicího uzlu
- *C\_KRABICKA* - model krabičky
- *D\_MERICI\_RIDICI\_UZEL* - fotografie měřicího a řídicího uzlu
- *E\_VIZUALIZACE* - zdrojový kód a obrázky vizualizace